Model Recommendation with Virtual Probes for Egocentric Hand Detection

Cheng Li
Tsinghua University
Beijing, China
chengli.thu@gmail.com

Kris M. Kitani
Carnegie Mellon University
Pittsburgh, PA, USA
kkitani@cs.cmu.edu

Abstract

Egocentric cameras can be used to benefit such tasks as analyzing fine motor skills, recognizing gestures and learning about hand-object manipulation. To enable such technology, we believe that the hands must detected on the pixel-level to gain important information about the shape of the hands and fingers. We show that the problem of pixel-wise hand detection can be effectively solved, by posing the problem as a model recommendation task. As such, the goal of a recommendation system is to recommend the n-best hand detectors based on the probe set – a small amount of labeled data from the test distribution. This requirement of a probe set is a serious limitation in many applications, such as ego-centric hand detection, where the test distribution may be continually changing. To address this limitation, we propose the use of virtual probes which can be automatically extracted from the test distribution. The key idea is that many features, such as the color distribution or relative performance between two detectors, can be used as a proxy to the probe set. In our experiments we show that the recommendation paradigm is well-equipped to handle complex changes in the appearance of the hands in first-person vision. In particular, we show how our system is able to generalize to new scenarios by testing our model across multiple users.

1. Introduction

Egocentric videos extracted from wearable cameras (*e.g.*, mounted on a person's head, chest or shoulder) can provide an up-close view of the human hands and their interactions with the physical world. We believe that this unique viewing perspective can be used to advance such tasks as analyzing fine motor skills, recognizing gestures and learning about hand-object manipulation. To enable such technology, we also believe that the hands must be detected on the pixel-level to gain important information about the shape of the hands and fingers. Therefore, we aim to extend the state-of-the-art in egocentric hand detection to provide a more stable pixel-resolution detection of hand regions. In particular, we will show that the problem of pixel-wise hand detection can be effectively solved by posing the problem as a model recommendation task. The role of our proposed recommendation system is to suggest the n-best hand detectors based on information extracted from the test image.

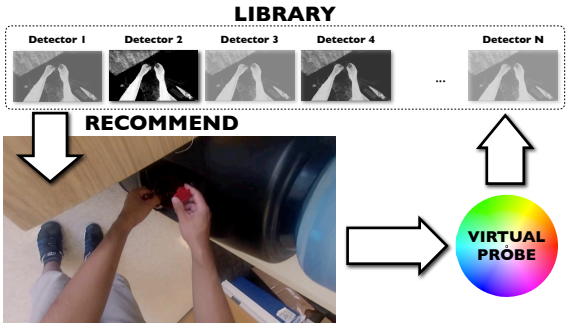

Figure 1. Ego-centric hand detection as a model recommendation task. Virtual probe features are extracted at test time to recommend the best detector performance.

In a typical recommendation task, information from the test distribution is acquired through a small amount of labeled data from the test distribution called the *probe set*. In the original context of recommendation systems such as movie recommendation, that probe set can be easily obtained by allowing a specific user to rank a small set of movies, safely assuming that the preferences of the user will not change drastically over time. In the case of egocentric hand detection, the probe set would amount to a small number of labeled pixels provided by the user. Based on this information, the recommendation system could return a set of scene appropriate detectors. However, in the case of a first-person camera where the user is constantly moving, the test distribution (*i.e.*, appearance of the hands, imaging conditions) is constantly undergoing change, rendering the initial probe set invalid. It would be impractical to update

the probe set dynamically, since this would require the user to label new pixels very time he moves.

A major difference between our egocentric hand detection scenario and movie recommendation is that we have access to a large amount of secondary information about the test subject (*i.e.*, the test image). While we do not have direct information about hand regions, such information about the brightness of the scene, objects in the scene and the structure of the scene can give us clues about the imaging conditions and help us infer what the hands might look like. Our claim is that this secondary source of information can be used to generate a *virtual probe set* to recommend the best detector.

Based on this observation, we propose to frame hand region detection for egocentric videos as a model recommendation task, where a *dynamic* virtual probe set is used to recommend a set of detectors for a dynamically changing test distribution. The contributions of this work are: (1) a novel dynamic classifier selection methodology applied to first-person hand detection and (2) a recommendation system framework that does not require a labeled *probe set*. In particular, we show that virtual probe features, namely global appearance and detector correlation, can be use to recommend the best detectors for test-time performance. Moreover, we show the effectiveness of our approach by showing improved performance on cross-user experiments for egocentric hand detection.

2. Previous Work

Previously the extraction of hands for egocentric vision has been posed as a figure-ground segmentation problem using motion cues [15, 5, 13]. One of the major advantages of motion-based hand detection approaches is that they are robust to a wide range of illumination and imaging conditions. A common feature among motion-based segmentation techniques is that they need to compute the dense [13] or sparse [15, 5] optical flow over a temporal window to discover the motion subspace spanned by foreground and background motion. A natural consequence of motion-based approaches is that they have a hard time segmenting regions for cases of extreme motion (*i.e.* no motion or large motion).

Traditional approaches to hand detection based on skin color [7] require that the statistics of the appearance are known in advance but have the benefit of being agnostic to motion. However, a problem arises when the distribution of hand skin color changes over time because a single skin color classifier cannot account for these changes. Previous work has explored the use of dynamic models to handle the gradual change in appearance [17] but may be prone to drifting when the change in the illumination is extreme.

In the case of an egocentric camera, the camera is mobile and unconstrained (*i.e.* the user can walk indoors or outdoors), so it is important that the hands can be detected under a wide range of imaging conditions and also be robust to extreme motion. In a recent work, Li and Kitani [9] have shown that hands can be detected at the pixel-level for egocentric videos under different imaging conditions using only appearance. In their framework, a global color histogram was used as a proxy feature to find a hand region detector trained under similar imaging conditions. However, since a color histogram folds both the appearance and illumination conditions onto a single feature space, it has difficulty generalizing to new scenes with similar imaging conditions but with different appearance (*e.g.* hand under sunlight in a previously unseen environment).

Matikainen *et al*. [10] has shown that the recommendation system paradigm can be very effective for automated visual cognition tasks such as action recognition, when only a small amount of training data is available. However, in their scenario the test distribution was assumed to be static. As we have described above this is not the case for egocentric hand detection where the test distribution is undergoing constant change. We present a probe-free recommendation approach over a dynamically changing test distribution.

A recommendation system approach differs from a standard supervised detection paradigm in that the detector is given the ability to adaptively change its parameters based on features extracted from the test distribution. Similar ideas have been investigated in areas of domain adaptation [14], transductive learning [6], kernel density ratio estimation[18], multi-task learning [2] and list/sequence optimization [4]. While a full comparison of differing approaches is outside the scope of this paper, we believe that leveraging the test distribution as part of the detection process is a powerful approach when applied to many vision tasks.

3. Preliminaries

Under our recommendation system paradigm, it is necessary to define the (1) set of models, (2) set of tasks, (3) a score (or ratings) matrix, (4) a set of probe models and (5) the recommender system.

The *set of tasks* is a large set of labeled data $\{\boldsymbol{x}_n, \boldsymbol{y}_n\}_{n=1}^{N}$, where $\boldsymbol{x}$ is the data and $\boldsymbol{y}$ is the label. In our scenario, each data sample $\boldsymbol{x}$ is a color image and $\boldsymbol{y}$ is a pixel-wise labeling of the hand regions.

The *set of models* is a large pool of functions $\{f_m(\boldsymbol{x})\}_{m=1}^{M}$, where each function generates a scalar value response for each task. In our scenario, a model is a random forest regressor that predicts a value between 0 and 1, where the regressor has been trained on various subsets of an egocentric hand dataset using a specific set of image features (*e.g.*, color descriptors, texture descriptors). However, there is no constraint on the type of classifier or input features, as long as the features can be extracted from the test set and

Figure 2. Sample images of the ego-centric videos used for evaluation.

models share a common output space.

The *score matrix* $\mathbf{R} \in \mathbb{R}^{M \times N}$ consists of the score $r_{mn} = f_m(\boldsymbol{x}_n)$ of the m-th model evaluated on the data of the n-th task . The rows of the score matrix are indexed by the models and the columns are indexed by the tasks. In our scenario each element of the matrix contains the $0-1$ loss computed by testing a regressor on a labeled image.

The *set of probe models* is a small number of models, which are used to evaluate a small group of labeled data from the test distribution (this small group of labeled data is sometimes called the 'training data' but we will call it the *probe data* to avoid confusion). The set of probe models $f_p(\boldsymbol{x})$ is typically a subset of the collection of models. Later we will introduce a disjoint set of models called the virtual probe features as a proxy to this set of probe models.

The role of a *recommendation system* is to use the response of the probe models on the probe data in order to recommend the best model for evaluating the test set. The recommendation system defines a mapping from probe responses to a model.

4. Detecting Pixel-wise Hand Regions

Due to the dynamic nature of first-person vision, we would like to adaptively select an appropriate hand model for every incoming image frame. In the following, we explain our use of virtual proxy features which can be used in the place of a probe set, thereby allowing the model to retain the predictive capabilities of a recommendation system without the restriction of a labeled probe data set.

4.1. Virtual Probe Features

Since we do not have access to labeled probe data, we would like to identify a set of proxy models or features $\{\hat{f}_v(\boldsymbol{x})\}_{v=1}^{V}$ to help define a mapping from the test image to a list of high-performance detectors. We call this set of proxy features as *virtual probe features*. We propose two types of virtual probe features: (1) global appearance features (extending the work of [9]) and (2) detector cross-correlation features.

Global appearance features such as a HSV histograms can be used as a proxy to the imaging conditions. Similarly, a large HOG [3] feature extracted of the entire image, similar to [16, 11] can be used to capture the structure of the scene. A full list of appearance-based virtual probe features are given in section 6.1 in Table 1.

In an effort to capture the predicted performance of detectors on the test image, we also propose the use of detector cross-correlation. For example, given a pair of detectors, where one is always better in bright scenes and the other is always better in low lit scenes, we can use the relative performance difference to infer the illumination of the scene. To compute the detector cross-correlation score, we first evaluate a base detector (*e.g.*, a mean detector) and a secondary detector on the test image to produce two response maps. The cross-correlation score is computed by aggregating the difference between the two response maps. Notice that this process does not require any labeled data since the cross-correlation score only encodes the relative performance of the two detectors. A similar representation was used in [10] for the internal representation of the score matrix but we are using it here as the virtual probe feature.

4.2. Augmented Score Matrix

Under the analogy of movie recommendation, a rankings database tell us how a particular user has ranked different movies. In the same way, our score matrix tells us how each model performed on each training image. Typically the recommendation system uses this score matrix to suggest a set of detector based on the response of the probe models. However, since we do not have access to a probe set and therefore cannot evaluate the probe models, we will use a set of virtual probe features as a proxy to the probe

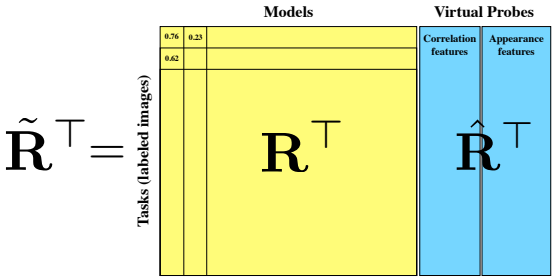

Figure 3. Structure of the augmented score matrix – a concatenation of models and virtual probe features on the training images.

models. This requires that we also store the response of the virtual probe features as part of the score matrix.

The standard score matrix is a large matrix $\mathbf{R} \in \mathbb{R}^{M \times N}$ of values indexed by a training image index n and a model index m. Each element $r_{mn} \in \mathbf{R}$ contains a scalar output of a model m when tested on training image n. In our experiments, r_{mn} is the normalized 0-1 loss computed from the thresholded output of a random tree regressor evaluated on a training image.

To incorporate the virtual proxy features, we augment the score matrix with virtual probe feature responses $\hat{r}_{vn}$ on the training data with the feature matrix $\hat{\mathbf{R}} \in \mathbb{R}^{V \times N}$, where V is the number of virtual probes. Concatenating the score matrix with the features matrix, we obtain an augmented score matrix $\tilde{\mathbf{R}} \in \mathbb{R}^{(M+V) \times N}$. A visualization of the transpose of the augmented score matrix is given in Figure 3, where each row is indexed by training images n and the columns are indexed by models and virtual probe features.

4.3. Recommendation System

We would like our recommendation system to tell us the best performing hand detector given an arbitrary test image. In our scenario our recommendation system defines a mapping $h(\hat{\boldsymbol{r}}) \rightarrow \boldsymbol{r}$, from a set of probe feature values $\hat{\boldsymbol{r}} = \hat{\boldsymbol{f}}(\boldsymbol{x}^{\text{test}})$ extracted from a test image $\boldsymbol{x}^{\text{test}}$ to the estimated scores of the all models $\boldsymbol{r} = \boldsymbol{f}(\boldsymbol{x}^{\text{test}})$ on the test image. Following [10], we describe several strategies we evaluate for learning the recommendation (mapping) function $h(\hat{\boldsymbol{r}})$.

4.3.1 Factorization

Matrix factorization can be used to discover a latent low dimensional representation of the augmented score matrix. We use non-negative matrix factorization [8] to decompose the augmented score matrix, $\tilde{\mathbf{R}} = \tilde{\mathbf{U}}^{\top}\tilde{\mathbf{W}}$, where $\mathbf{U}$ is a non-negative $(M+V) \times K$ matrix and $\tilde{\mathbf{W}}$ a non-negative $K \times N$ matrix. $\tilde{\mathbf{U}}$ spans a K dimensional imaging subspace and $\mathbf{W}$ describes each of the N training images as a K-dimensional mixture vector. Recall that the rows of the augmented score matrix can be separated into the V virtual probe responses and M model responses. At test time, the virtual probe features of the test image $\hat{\boldsymbol{r}}$ can be used to solve for the weight vector $\boldsymbol{\theta}$ of the sub-matrix $\hat{\mathbf{U}}$ to satisfy

$$\hat{\mathbf{U}}^{\top}\boldsymbol{\theta} = \hat{\boldsymbol{r}}. \tag{1}$$

Then to predict the models response on the test image, we solve $\boldsymbol{r} = \mathbf{U}^{\top}\boldsymbol{\theta}$.

4.3.2 Sparse Coding

A sparsity prior can also be enforced on the matrix $\hat{\boldsymbol{R}}$ via a sparse weight vector $\boldsymbol{\alpha}$, which is used to select a sparse set of virtual probe features to span the imaging conditions. An optimal sparse weight vector is computed by

$$\boldsymbol{\alpha}^{*} = \arg\min_{\boldsymbol{\alpha}} \|\hat{\boldsymbol{r}} - \hat{\mathbf{R}}\boldsymbol{\alpha}\|_2^2 + \tau\|\boldsymbol{\alpha}\|_1, \tag{2}$$

where $\hat{\boldsymbol{r}}$ are the responses of the virtual probe features on the test image, $\hat{\mathbf{R}}$ are the rows of the augmented score matrix corresponding to the virtual probe features, and α is the vector of weights for the sparse reconstruction. τ is the sparsity hyper-parameter. Once $\boldsymbol{\alpha}^{*}$ has been computed, the predicted model responses $\boldsymbol{r}$ can be computed simply as the weighted combination of columns of $\mathbf{R}$.

4.3.3 Nearest Neighbor

Another simple way to map a set of virtual probe features $\hat{\boldsymbol{r}}$ to model scores $\boldsymbol{r}$, is to treat the virtual probe features as a direct index into the augmented score matrix. At test time, we extract the virtual probe features and then find the training image with the most similar virtual probe feature response distribution using a nearest neighbor search. This is the same approach used in [9], where a HSV color histogram was used as an index to find the nearest image frame in the database and then used a set of classifiers associated with that image on the test image. It was shown that this feature can be effective when the dataset is always a superset of the test images.

4.3.4 Non-linear Regression

Since our augmented score matrix is dense (no missing data) we can take a step further and attempt to learn a non-linear mapping between virtual probe features $\hat{\boldsymbol{r}}$ and model scores $\boldsymbol{r}$ with a non-linear regressor $g(\hat{\boldsymbol{r}}) \rightarrow \boldsymbol{r}$. In our experiments we evaluate a random forest regressor to estimate test time model scores.

5. Hand Region Segmentation

While our proposed pixel-level detection of hand regions is robust in various scenarios, it also important to ensure

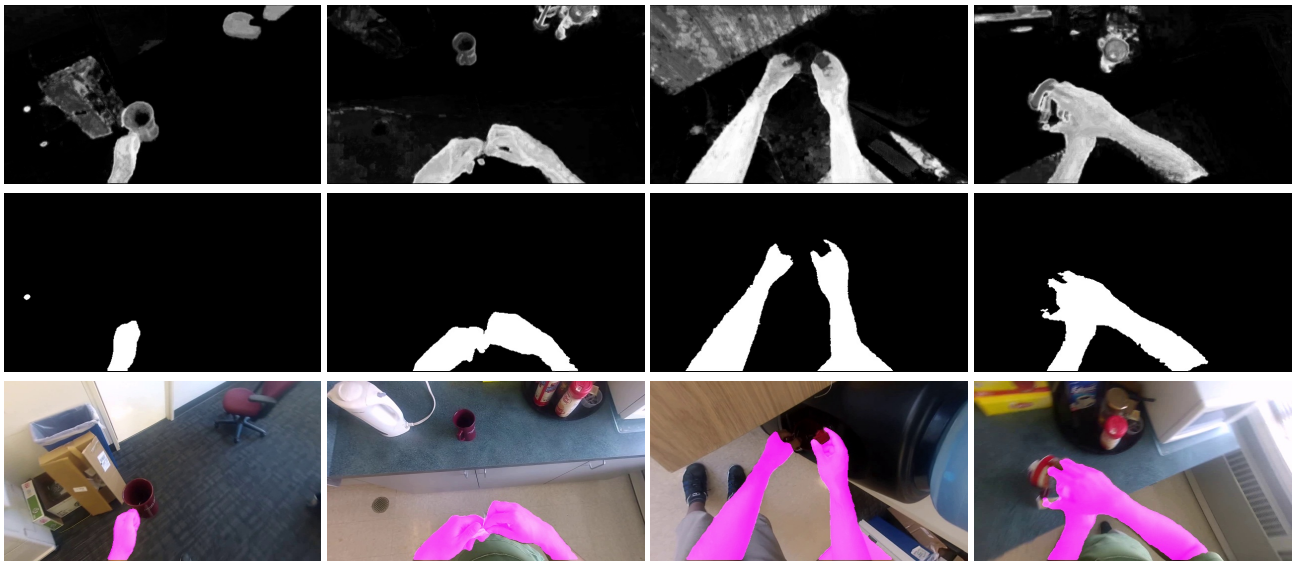

Figure 4. Hand region detection results: per-pixel likelihood (**top**), segmentation (**middle**) and final result (**bottom**).

global consistency between pixel-wise detections using top-down cues. As in many segmentation techniques, we formulate the task of hand region contour segmentation as an energy minimization problem [1] over super-pixel regions [13, 15, 5]. Our spatio-temporal super-pixel graph aims to extract consistent regions by modeling temporal smoothness, spatial smoothness and a spatial prior.

Our energy function is defined as

$$\begin{aligned}\log\ p(\boldsymbol{L}|\boldsymbol{x}) = &\sum_i \phi_i^{\text{like}} l_i + \sum_i \theta \phi_i^{\text{pos}} l_i \\ &+ \sum_{ij} \lambda \phi_{ij}^{\text{spat}} \Big[2 l_i l_j - (l_i + l_j) + 1\Big] \\ &+ \sum_{ik} \nu \phi_{ik}^{\text{temp}} \Big[2 l_i l_k - (l_i + l_k) + 1\Big],\end{aligned} \tag{3}$$

where i indexes the superpixels at time t, j indexes all spatially neighboring super-pixel at time t, and k indexes all temporally neighboring superpixels within a finite temporal window. An illustration of the spatial and temporal potentials are given in Figure 5. The optimization yields segmentation results visualized in Figure 4.

The unary likelihood potential ϕ^{like} is defined as the log odds, the mean hand likelihood of all pixels within a super-pixel belonging to the foreground class divided by the likelihood of the background class. Likewise the unary position prior ϕ^{pos} is computed from the mean position likelihood of pixels (computed from a 2D Gaussian centered at the centroid of the nearest connected component). The spatial binary potentials ϕ_{ij}^{spat} is defined as the probability of the mean LAB values of super-pixel j, modeled by a Gaussian centered at the mean of super-pixel i. Following [19],

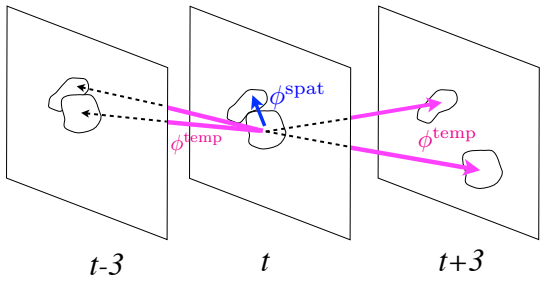

Figure 5. Visualization of the binary potentials of our spatio-temporal graph used for segmentation.

the temporal binary potential ϕ_{ik}^{temp} is an indicator function that is unity when two super-pixels overlap, where overlap is computed at the spatial intersection of two super-pixels i and k, after super-pixel i has been shifted according to the average optical flow between time t and $t + w$ (the time index of super-pixel k). We use a temporal window of $\pm$ 6.

6. Experimental Evaluation

We use three publicly available ego-centric datasets to evaluate our proposed hand detection algorithm. The CMU EDSH dataset contains three sequences, containing over 400 pixel-level image labels [9]. As this dataset was created for hands under varying illumination, the hands of one person is recorded under various imaging conditions but does not contain a wide range of actions. We use videos from 6 different subjects from the UCI dataset [12], where users are engaged in various activities of daily living (ADL). This

dataset is the most challenging, as video is taken by a chest worn camera (fingers are harder to detect) and taken in a wide range of indoor imaging conditions. We also used the Georgia Tech egocentric activities (GTEA) dataset [5] to test our segmentation algorithm.

For all of our experiments, we use the local patch-based random forest regressor used in [9] as our base detector using LAB, HSV and BRIEF features.

6.1. Evaluating Probe Features

In this experiment we are interested in the ability of virtual probe features (global appearance features and detector cross-correlation features) to improve the performance of hand detection. We tested 20 different variations of virtual probe combinations over the CMU EDSH dataset and the UCI ADL dataset. The set of models for the CMU EDSH dataset were generated from the EDSH1 video, by clustering images by their HSV histogram and training a separate model for each cluster. We used the same procedure for the UCI ADL dataset to generate a pool of models. For the EDSH data the average of the top 19 models were used to compute the F-measure and in the ADL dataset the weighted average of the top 5 models were used to compute the F-measure. NMF was used as the recommendation technique. The results are summarized in Table 1.

The baseline method is a single detector trained on all the training data. This baseline represents a model without any concept of model recommendation and therefore has no virtual probe features. Since the model is forced to represent all hand features with a single model it yields the lowest performance.

First, we evaluated HSV color histograms and global HOG [3] over a variety of spatial bins as a virtual probe feature. The HSV histogram is 64d ($4 \times 4 \times 4$) and the HOG template is 81d. The F-measures of the appearance features are given to the left of the slash symbol in Table 1. We can see from the distribution of scores in bold, that the HSV-based virtual probes obtain the best performance for the majority of datasets. Although in 4 of the 8 ADL datasets the HOG feature also generates the best score. This indicates that both the color of the scene and the structure of the scene are helpful in determining the best selection of models.

Second, we evaluated cross-correlation features. We treat the output of a mean model f_0 as 'true' and compute the 0-1 loss of another model m with respect to the output of the mean model. For each test of the CMU EDSH dataset, the number of models was $M = 242$ (including the mean model) and therefore has $M - 1$ cross-correlation features. Each test of the UCI ADL dataset utilized 180 models. The F-measure obtained by the addition of the cross-correlation feature is given to the right of the slash symbol in Table 1. We see from the right-most column that the cross-correlation feature improves performance on average. This indicates that the cross-correlation feature is indeed encoding useful information about performance on the test distribution.

Table 1. Evaluating different variations of probe features. Left of slash is the F-measure with only global feature and the right of slash performance combined with cross-correlation features.

Virtual Probe	EDSH2	EDSH-K	ADL (avg.)
No Probe	0.788	0.806	0.265
HSV (1)	0.821 / 0.844	**0.849** / **0.822**	**0.302** /0.351
HSV (top/bot)	0.822 / 0.847	0.846 / **0.822**	0.229 /0.348
HSV (2 by 2)	**0.825** / 0.845	0.839 / **0.822**	0.212 /0.309
HSV (3 by 3)	0.824 / **0.848**	0.837 / 0.82	0.215 /0.342
HSV (1+3)	0.820 / 0.846	0.841 / 0.823	0.264 /0.331
HoG (1)	0.752 / 0.836	0.801 / 0.814	0.285 / **0.358**
HoG (top/bot)	0.768 / 0.838	0.807 / 0.811	0.235 /0.339
HoG (2 by 2)	0.777 / 0.843	0.807 / 0.813	0.200 /0.325
HoG (3 by 3)	0.774 / 0.836	0.808 / 0.814	0.200 /0.307
Corr. only	/ 0.843	/ 0.810	/0.339

Table 2. Evaluating recommendation strategies.

Recommendation	EDSH2	EDSH-K	ADL_AVG
NMF	0.834	0.811	0.322
SC	0.781	0.812	0.252
KNN	0.843	0.805	**0.384**
RF	**0.848**	**0.825**	0.357
No Probe (single)	0.765	0.800	0.265
Sparse Feature [9]	0.781	0.808	0.346

6.2. Comparing Recommendation Strategies

We now compare the four recommendation strategies explained in section 4.3 and two baseline models. For each recommendation experiment, we use the same parameters as the previous experiment but using the best combination of virtual probe features (*i.e.* the best HSV, best HOG and cross-correlation feature combination).

Table 2 shows that our recommendation approach beats the state-of-the-art detection of [9]. Furthermore, we observe that the non-linear models (NN regression and RF regression) perform better than the linear factorization (NMF and SC) models on both datasets. Non-linear models have the benefit of capturing more complex mappings between the probe features and the unobserved features. However, non-linear models also have two drawbacks. First, a large number of virtual features increases the possibility of overfitting to the data in the score matrix. Second, in the case of the RF model, the mapping from virtual probes to model scores is expensive, since a single RF model is trained for each entry of the score matrix. We will analyze and evaluate these characteristics in the next section 6.3.

6.3. Minimizing Correlation Feature Usage

In the previous experiments, many cross-correlation features were used as virtual probe features. However, since

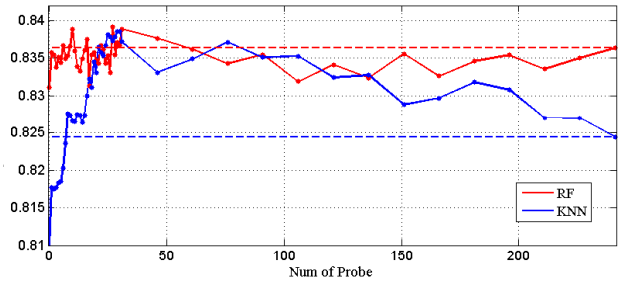

Figure 6. Performance versus number of correlation probe features. Only a small number (around 10) of probes are necessary for robust and efficient performance.

each cross-correlation requires the evaluation of the entire test image, using a large number of cross-correlation features can be expensive and not practical for real-time applications that require a fast response time. Also as mentioned previously, a large number of probe features can also cause the non-linear recommendation schemes to over-fit to the data. In this section, we examine the tradeoff between computation time and performance, by varying the number of virtual cross-correlation probe features.

We plot the change in performance on the EDSH dataset by increasing the number of cross-correlation probe features. The number of global appearance probe features (combination of HSV and HOG features) remains constant throughout. When the number of probes is 0, only the global appearance features are being used. Figure 6 shows the results for the top performing non-linear recommendation strategies using the random forest (RF) and k-nearest neighbors (KNN). The dotted lines indicate the performance when all 241 cross-correlation features are used.

Although we expected the RF recommendation approach to overfit to the data, we observed that the RF is relatively stable. We believe this robustness comes from the built-in random features selection process of the RF model. When the set of models is smaller than the number of pixels in the test image, the RF model will be the most efficient approach. It is interesting to note that the simple KNN approach can obtain the same level of performance as the RF approach when about 30 cross-correlation features are used but it also quickly overfits as more features are introduced.

6.4. Evaluating Potentials for Post-Processing

In our segmentation step we introduced an energy function based on three potential functions and a label bias parameter. Table 3 shows the results of ablative analysis by removing one potential at a time. F-measures values are given for the EDSH dataset and GTEA dataset. We observed that the temporal potential provided the greatest contribution, especially on the EDSH dataset which contains

Table 3. Time-Space MRF with one parameter fixed in zero

	EDSH2	EDSH-K	GT-T	GT-P
All parameters	0.828	0.883	0.911	0.800
No position prior ($\theta = 0$)	0.812	0.874	0.898	0.791
No temporal smoothing ($\nu = 0$)	0.806	0.872	0.897	0.784
No spatial smoothing ($\lambda = 0$)	0.827	0.863	0.894	0.784
All parameters (keep 3 contours)	0.828	0.886	0.942	0.825

Figure 7. Segmentation results on the GTEA dataset.

Table 4. Cross-User Performance on the UCI ADL dataset. Leave-one-out style training where probe includes global appearance and detector cross-correlation features.

Probe	User1	User2	User3	User4	User5	User6	avg.
No probe	0.204	0.209	0.326	0.172	**0.342**	0.337	0.265
NMF	0.199	0.291	0.572	0.169	0.288	0.413	0.322
SC	0.186	0.321	0.386	0.135	0.068	0.418	0.252
KNN	0.254	**0.414**	0.569	**0.358**	0.232	**0.480**	**0.384**
RF	**0.274**	0.298	**0.650**	0.232	0.327	0.362	0.357

large degrees of ego-motion, where the user is walking for most of the sequence. The best performance was achieved by using all potentials. We also obtain a small improvement when we use a simple post-process step to keep only the top 3 largest contours. Examples of segmentation from the GTEA dataset are given in Figure 7 and results for the EDSH dataset are given in Figure 4.

6.5. Cross-user Performance

Many first-person vision systems can be personalized to a single user since the camera will only be used for one person. However, in other applications, it may not be possible to gather labeled pixel-wise ground truth data of a specific user. Therefore, we would like to know the performance of our proposed approach when we are not given any training data for the test user. For this experiment we use only the ADL dataset, since the EDSH dataset only contains data for a single person.

Table 4 shows the performance of cross-user performance on the UCI ADL dataset, where training data from 5 users are tested on single held out user in a leave-one-out style rotation of the data. We use the same no probe single detector baseline to show how our recommendation approach can be used to adapt to new users in various lighting conditions. A sample of the final output is given in Figure 8. The absolute scores and segmentations (Figure 9) are far from perfect. This shows the challenging nature of detecting hands in real life scenarios especially in very dim lit scenes where it is hard to detect skin texture.

Figure 8. Sample results on the UCI ADL dataset.

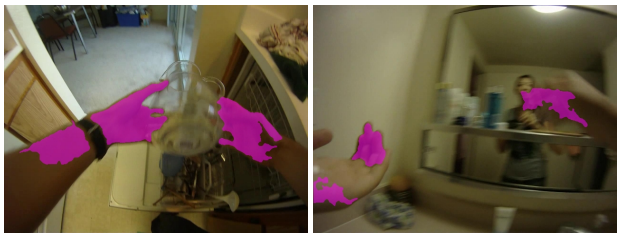

Figure 9. Incomplete detections.

7. Conclusion

In this work it was our aim to extend the state-of-the-art in egocentric hand detection to provide a more stable pixel-resolution detection of hand regions. In particular, we showed that the problem of pixel-wise hand detection can be effectively solved, by posing the problem as a model recommendation task. Through quantitative analysis we showed that our proposed approach is able to retrieve the best hand detectors based on global appearance features and cross-correlation feature extracted from the test image. We also evaluated the role of proper post-processing and showed that pixel-level detections should be verified by a top-down post-processing step to ensure certain global properties about the hands. In our experiments we showed robust hand detection by testing our model across multiple users and showed that our proposed approach attains state-of-the-art performance.

Acknowledgements

We thank Pyry Matikainen for discussions regarding model recommendation and the initial inspiration for using detector cross-correlation. This research was supported in part by NSF QoLT ERC EEEC-0540865. Li was also supported by the Sparks Program at Tsinghua University and Prof. Xiaoou Tang from CUHK.

References

[1] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *PAMI*, 26(9):1124–1137, 2004. 5

[2] R. Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997. 2

[3] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3, 6

[4] D. Dey, T. Liu, M. Hebert, and J. A. Bagnell. Contextual sequence prediction via submodular function optimization. In *Robotics Science and Systems*, 2012. 2

[5] A. Fathi, X. Ren, and J. Rehg. Learning to recognize objects in egocentric activities. In *CVPR*, 2011. 2, 5, 6

[6] T. Joachims. Transductive inference for text classification using support vector machines. In *ICML*, 1999. 2

[7] M. Jones and J. Rehg. Statistical color models with application to skin detection. In *CVPR*, 1999. 2

[8] J. Kim and H. Park. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *International Conference on Data Mining*, 2008. 4

[9] C. Li and K. M. Kitani. Pixel-level hand detection for ego-centric videos. In *CVPR*, 2013. 2, 3, 4, 5, 6

[10] P. Matikainen, R. Sukthankar, and M. Hebert. Model recommendation for action recognition. In *CVPR*, 2012. 2, 3, 4

[11] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 3

[12] H. Pirsiavash and D. Ramanan. Detecting activities of daily living in first-person camera views. In *CVPR*, 2012. 5

[13] X. Ren and C. Gu. Figure-ground segmentation improves handled object recognition in egocentric video. In *CVPR*, 2010. 2, 5

[14] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *ECCV*. 2010. 2

[15] Y. Sheikh, O. Javed, and T. Kanade. Background subtraction for freely moving cameras. In *ICCV*, 2009. 2, 5

[16] A. Shrivastava, T. Malisiewicz, A. Gupta, and A. A. Efros. Data-driven visual similarity for cross-domain image matching. *SIGGRAPH ASIA*, 30(6), 2011. 3

[17] L. Sigal, S. Sclaroff, and V. Athitsos. Skin color-based video segmentation under time-varying illumination. *PAMI*, 26(7):862–877, 2004. 2

[18] M. Sugiyama, T. Kanamori, T. Suzuki, S. Hido, J. Sese, I. Takeuchi, and L. Wang. A density-ratio framework for statistical data processing. *IPSJ Transactions on Computer Vision and Applications*, 1:183–208, 2009. 2

[19] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple hypothesis video segmentation from superpixel flows. In *ECCV*. 2010. 5